# Robust and automated solution for correcting hotspots locally using cost-function based OPC solver

Carl Babcock[a], Dongok Yang[b], Sarah McGowan[a], Jun Ye[c] , Bo Yan[c], Jianhong Qiu[c], Stanislas Baron[c], Taksh Pandey[c], Sanjay Kapasi[c], Chris Aquino[c]
[a]GLOBALFOUNDRIES, Santa Clara, CA 95054; [b]GLOBALFOUNDRIES, Singapore 738406;
[c]ASML Brion, Santa Clara, CA 95054, USA

## ABSTRACT

In previous work[1], we introduced a new technology called Flexible Mask Optimization (FMO) that was successfully used for localized OPC correction. OPC/RET techniques such as model-based assist feature and process-window-based OPC solvers have become essential for addressing critical patterning issues at 2x and lower technology nodes. With an FMO flow, critical patterns were identified, classified and corrected in localized areas only, using advanced techniques. One challenge with this flow is that once the hotspots are identified, a user still has to come up with OPC solutions to address the hotspots. This process can be cumbersome and time consuming as different types of hotspots with new designs may require different recipes, causing delays to tapeout. What is required is a robust, powerful and automated OPC technique that can handle various types of hotspots, so an automatic hotspot correction flow can be established.
In this work, we introduce a new cost-function-based OPC technique called Co-optimization OPC that can be used to correct various types of hotspots with minimum tuning effort. In this approach, the OPC solver simultaneously solves for all the segments in a patch including main and sub-resolution assist features (SRAF), applying additional user-defined cost function constraints such as MEEF, PV band, MRC and SRAF printability. Unlike conventional OPC solvers, Co-optimization solvers can also move and grow SRAFs, which further improves the process window. The key benefit of the Co-optimization OPC solution is that it can be used in a standard recipe to resolve many different hotspots encountered across various designs for a given layer.
In this study, we demonstrate that Co-optimization OPC can be successfully used to address various types of hotspots across designs for selected 2x nm node line/space layers, as an example. These layers have been particularly challenging as they use single-exposure lithography with k1 around 0.3. Aggressive RET solutions are required to address the patterning challenges for this layer. Finally, we will report on implementation of the Co-Optimization OPC Recipe within the FMO framework for hotspot correction.

**Keywords:** Hotspot, OPC, Optical Proximity Correction, FMO, Flexible Mask Optimization, Co-Opt, Co-Optimization

## 1. INTRODUCTION

Leading edge nodes in semiconductor manufacturing present many new challenging patterns for lithography and Optical Proximity Correction (OPC). When this OPC challenge is combined with the requirement of fast turn-around time for mask data prep, tight tolerances for OPC printability verification, and a rapid flow of new designs through mask data prep, we see a need for the capability to combine established, fast, full-chip OPC with a selectively applied second advanced OPC that spends a longer run time to perform a more elaborate correction method on selected small areas. Another way of thinking about this approach is to call the second pass of OPC "repair OPC", since it works to repair hotspots from the first-pass OPC. The key requirement for the second OPC technology is that it should be robust and powerful to address various types of hotspots encountered for a given layer in production environment. To put his in production flow, an automated framework is required.

In previous work[1], it was demonstrated that the Flexible mask optimization (FMO) framework can be used to address hotspots locally in a production environment. However, this flow still relies on users to come up with customized patches to fix the issues thereby increasing the final time to solution. In this study, we introduce a new and advanced OPC solution called Co-optimization OPC that can be used to resolve many types of critical hotspots. This solution is ideal to be used within the FMO framework. It makes it possible to automate the full flow, to reduce the final time to solution, and to achieve the desired process window and a defect-free tapeout. In this paper, we first describe the FMO

Design-Process-Technology Co-optimization for Manufacturability VIII, edited by John L. Sturtevant, Luigi Capodieci, Proc. of SPIE Vol. 9053, 90530Z · © 2014 SPIE
CCC code: 0277-786X/14/$18 · doi: 10.1117/12.2046662

flow, then provide details about the Co-optimization solver and our implementation of these two techniques with real examples from 2x nm node layers.

## 2. DESCRIPTION OF FLEXIBLE MASK OPTIMIZATION PROCESS FLOW

### 2.1 Basic framework

The process starts by running a standard production SRAF and OPC recipe. This is followed by the standard LMC (Litho Manufacturability Check) recipe which checks for printability spec. violations, or "hotspots". Usually there will be no critical hotspot, but occasionally LMC detects one or more critical hotspots, i.e. printability check violations. The LMC recipe may have a provision for generating warning defects which need not be improved through the FMO application.

In the case that there are one or more hotspots as shown in the illustration, a list of the hotspot co-ordinates is compiled. An OPC engineer will then be able to run tests of various OPC enhancements to arrive at a recipe enhancement that resolves the hotspot and results in a clean LMC check in the small window area around the hotspot. At this point the enhanced OPC recipe together with the hotspot window coordinates are entered in to the FMO application, which runs the enhanced OPC inside each of the specified windows, and stitches these together with the existing OPC results outside of those windows. FMO uses model based boundary handling to ensure that the boundaries between the fixed area and the rest of the chip are free of MRC and LMC defects.

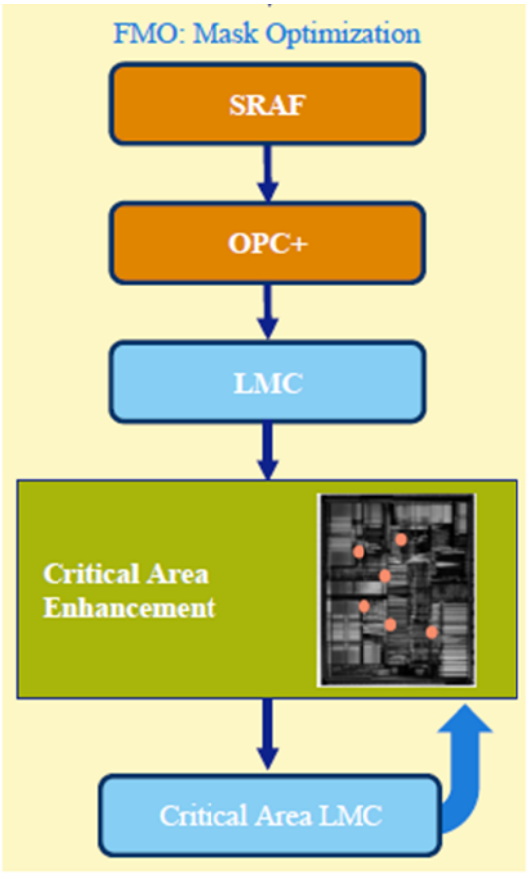


Figure 1. Flexible Mask Optimization (FMO) framework .

### 2.2 Issues with pre-existing FMO flow

While the current FMO flow offers a method to repair hotspots quickly, it leaves room for further improvement in efficiency. This is because it requires every hotspot to be reviewed by an OPC engineer, and then forces the OPC engineer to find an OPC recipe enhancement that will solve each hotspot. This consumes time and resources, which causes a risk of tapeout delays and limits the scalability of the FMO approach. Due to the tradeoff between engineering resource and mask tape out cycle time, current FMO can only focus on spec-violating defects. What is needed is a more powerful and flexible OPC approach that the FMO can apply on the hotspot areas automatically, and apply to all "warning defects" as well to further enhance the mask quality automatically. This eliminates the need to wait while an engineer examines individual hotspots and chooses ways to solve them. Because the FMO process will limit the use of the more powerful OPC to small areas at each hotspot, one can use a much more computationally intensive approach for this second-pass OPC. OPC techniques that could not have been used on a full chip due to excessive run time may in fact be used as the second-pass OPC in the FMO flow. For the purpose of this study we chose to apply new Co-Optimization OPC solution.

## 3. CO-OPTIMIZATION OPC

Co-Optimization OPC (also called Co-Opt) is one technique available in Brion's OPC+ package that can offer greater solving power in exchange for more computation time. Contrast this with single variable solver and local multi variable solver, referring to Figure 2. Single variable solver (SVS) adjusts an individual segment (e.g. #1) based only on its own simulated printed position during the current iteration. Local multi variable solver (LMVS) allows the user to specify the segments on a 2D structure that should be solved simultaneously (e.g. line-end segments, such as #2 – 6, 7 – 11, etc.) Co-optimization (Co-Opt) solver simultaneously solves for ALL the segments in a patch including main and SRAF (e.g. 1, 2-11, and a-d) with additional user defined cost function constraints such as MEEF, PV band, MRC or SRAF printability.

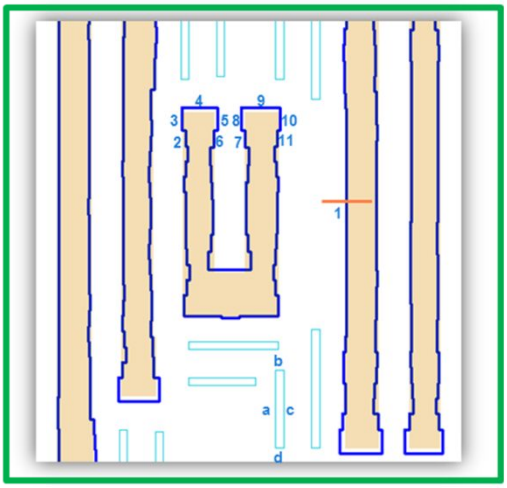


Figure 2 In local multivariable solver, the user-specified neighbor segments #1 - #11 are solved simultaneously to minimize their combined error. Co-Opt OPC solves all segments in the patch simultaneously.

Co-optimization OPC is an OPC solver that simultaneously solves for all the segments in a patch across conditions of focus, dose and mask bias, to minimize a user-specified cost function. Main and SRAF segments can be handled together and co-optimized. Different geometry types (e.g., line end, corner, run, etc.) can be handled by different weighting in cost functions. "Soft" constraints (MRC, SRAF printing, etc.) with relative weighting can be specified in the cost functions.

Dense evaluation points are defined on each segment, and the range of influence (non-zero Jacobian) can be controlled by user input and model ambit. User defined constraints for the cost function include MEEF, PV band width, MRC limits (mask width/space etc.) and SRAF printability, in additional to nominal printing error.

Co-optimization OPC can provide additional correction optimization beyond SVS or LMVS, with the aim of handling various types of OPC issues with minimal tuning efforts. It is part of the Tachyon OPC engine, so it can be implemented for production flows. Because of its longer runtime compared to SVS or LMVS, it makes sense to apply it in small areas of the layout, as is the case with the 2nd OPC stage of the FMO flow. Figure 3 shows examples of patterns where min width through Process Window was improved by the application of Co-Opt automatically, without additional engineering effort. In the first case (left), the minimum width in process window simulation was improved from 0.5 nm below the spec limit to 2.8 nm above the spec limit. In the second case (right), minimum width in process window simulation was improved from 1.6 nm below the spec limit to 12.4 nm above the spec limit.

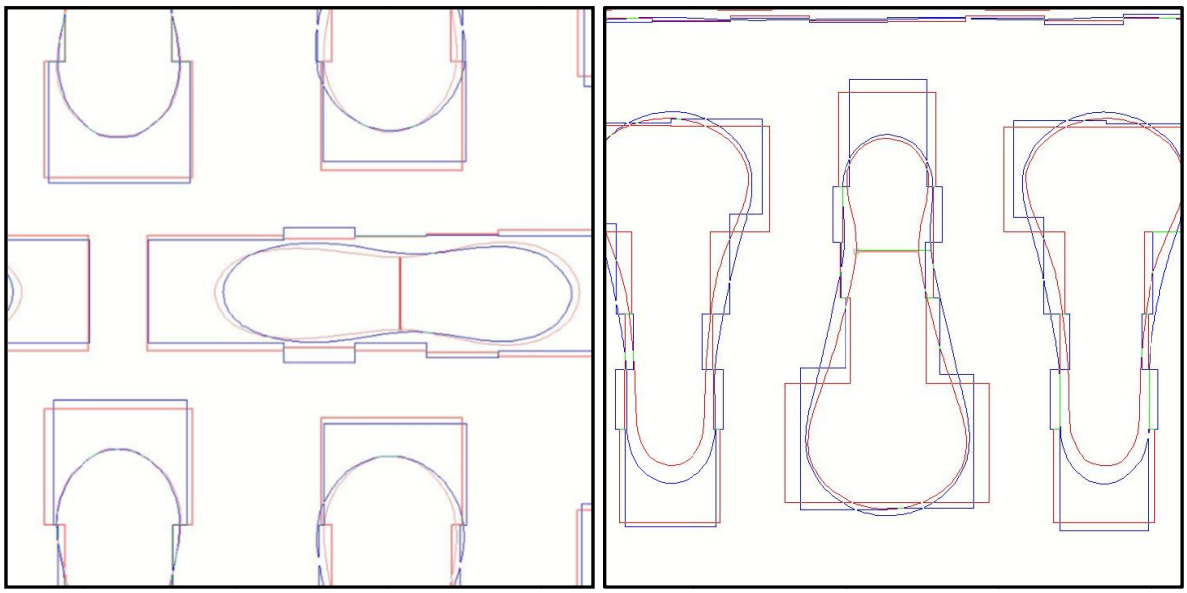

Figure 3 Example patterns where minimum width through process variation was improved by the application of Co-Opt (pink = SVS PWmin contour; blue = Co-Opt PWmin contour)

## 4. COMBINING CO-OPTIMIZATION INTO AUTOMATED FMO FLOW

For the purpose of this study, an automated FMO flow has been set up as shown in Figure 4.

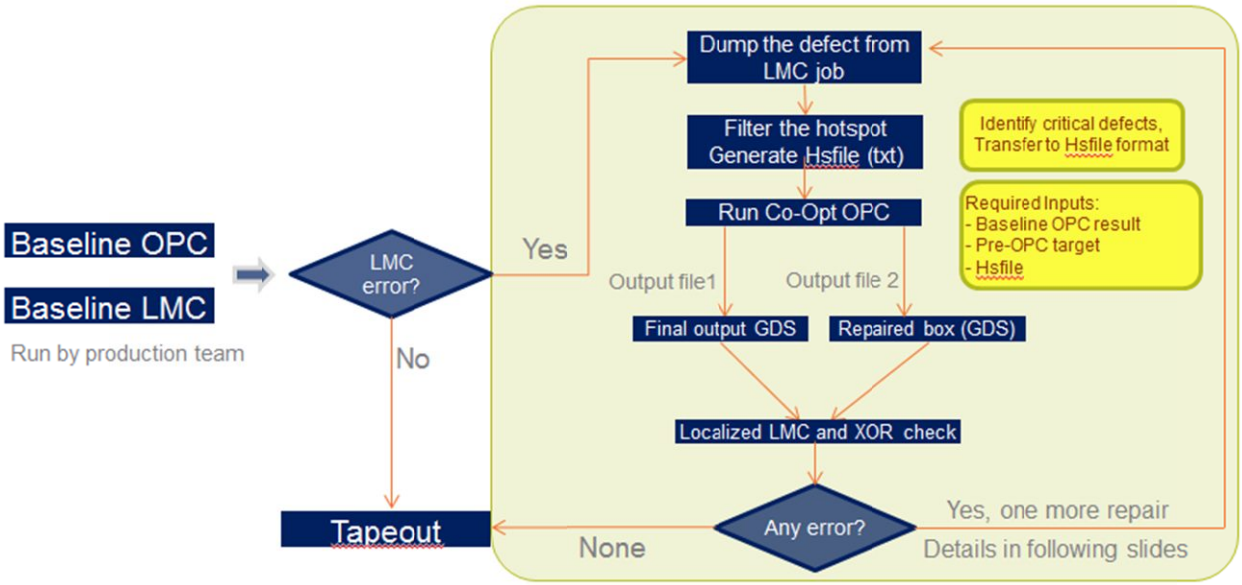


Figure 4 Automated FMO Flow set up to apply Co-Opt repair OPC to hotspots

Several large arrays of layout clips were assembled and used as test cases. After Baseline OPC, the defects were dumped into a hotspot file or "Hsfile."

The repair region size is specified by the user; Figure 5 presents sites for FMO repair showing the inner and outer repair boxes. The FMO OPC is performed inside the inner box. Outside of the outer box, there is no change to the Baseline OPC output. In the zone between inner and outer boxes, FMO uses model-based boundary handling to make any adjustments needed to resolve MRC errors and mismatch of the two different OPC solutions at the boundaries.

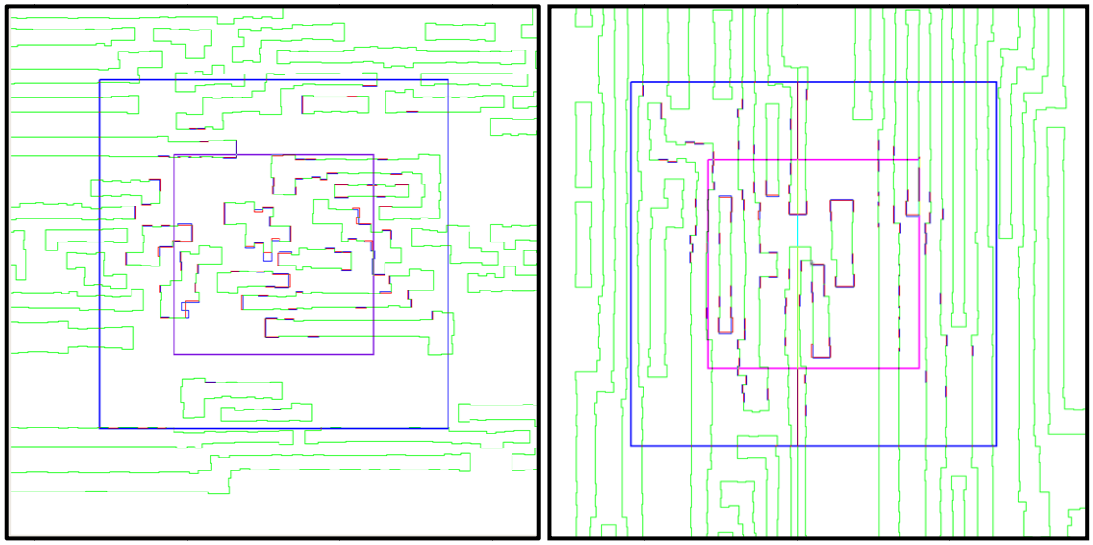

Figure 5 Repair regions indicated by inner and outer boxes on sample layouts

## 5. RESULTS OF AUTOMATED REPAIR ON TEST CASES

The FMO repair was tested on large arrays of clips of 2x nm node line/space layout. In order to test the capability of the automatic repair completely, we did 2 tests, the first test being a stress test with numerous defects. The defects to be repaired in this test were all Bridging defects and all the Necking defects. Both spec. defects and information (warning) defects are included. For the purpose of testing and verification of robustness of the solution, we wanted to test with large number of hotspots. In order to generate enough hotspots to sufficiently exercise the flow, the "baseline" OPC was detuned. After running the detuned baseline OPC, hotspots were identified and defect boxes generated; a sampling is shown in Figure 6, which only presents a small part of the full test chip, in order to make the defects visible in this study.

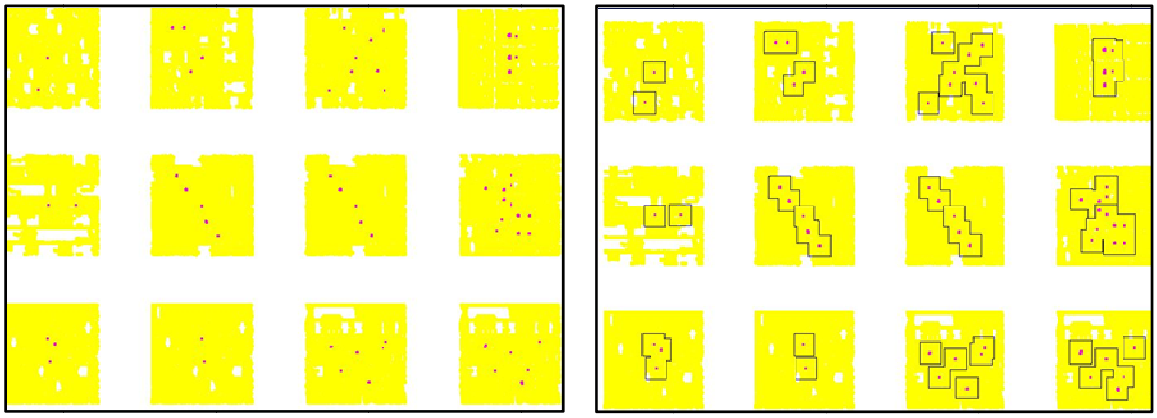

Figure 6 (Left) Hotspots remaining after "baseline" OPC run on test clips layout. (Right) repair boxes shown around hotspots

For some cases where MEEF is extremely high and contour CD is already close to the spec limit from baseline OPC, even a small variation in mask would create a new defect. To address these kinds of patterns and hotspots, the automatic FMO flow implements a guarding check in the form of a 2nd repair loop to avoid any degradation (Figure 7). This was again only encountered during the stress testing where large numbers of defects were found.

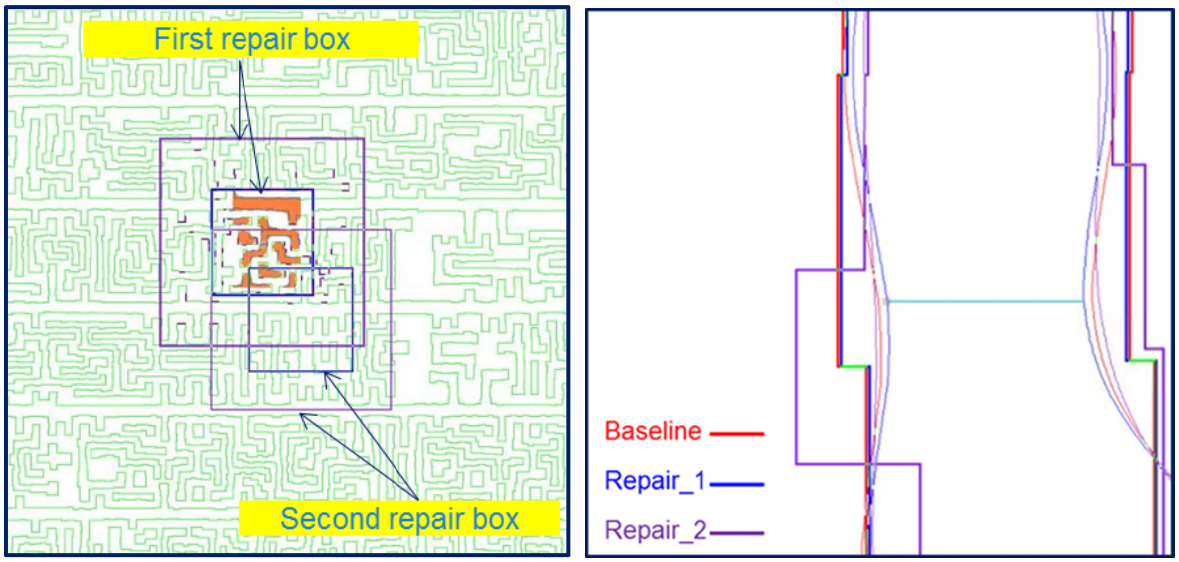


Figure 7 (left) 2nd repair for new defects. (Right) Mask and contour from Baseline, repair OPC 1 and repair OPC 2. At narrowest point of this site, Baseline contour CD is 1.6 nm above spec limit; Repair_1 contour CD is 1.3 nm below spec limit; finally Repair_2 contour width is 3.3 nm above spec limit.

After one iteration of repair OPC, only 2 hotspots remained in this region (Figure 8.) A second iteration of repair was performed, resulting in zero defects.

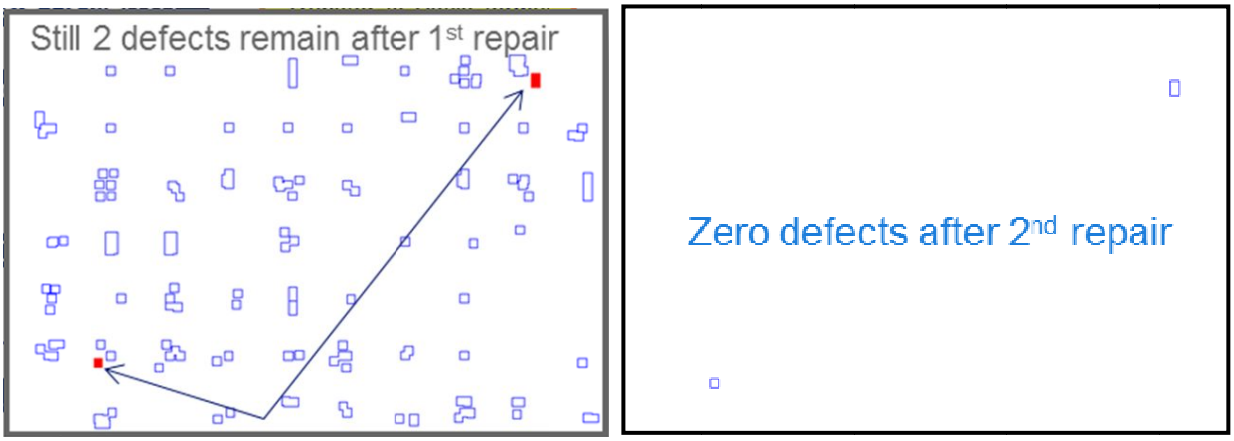


Figure 8 (Left) 2 hotspots remain after first repair OPC. (Right) No hotspots remain after 2nd repair.

After the second iteration of repair OPC, all critical hotspots had been repaired, which demonstrates that the flow worked well in a case with a large number of defects. (Table 1)

Table 1. Stress test with numerous defects

| *Layer* | *Defect number in Baseline OPC* | *Defect number after 1st repair* | *Defect number after second repair* |
|---|---|---|---|
| Layer 1 | 24062 | 81 | 0 |
| Layer 2 | 3821 | 51 | 0 |

The second test is a compatibility test within GLOBALFOUNDRIES tapeout environment. The test clips for this are hotspots collected from 2x nm tapeouts. The OPC recipe is the production recipe which is well tuned. The defects that are to be fixed are all bridging and necking defects. The result in (Table 2) demonstrates that the FMO automated repair flow can also yield clean results in production environment.

Table 2. Compatibility test in GLOBALFOUNDRIES production environment

| *Layer* | *Defect number in Baseline OPC* | *defect number after 1st repair* | *defect number after second repair* |
|---|---|---|---|
| Layer 1 | 203 | 2 | 0 |
| Layer 2 | 55 | 0 | N/A |

To estimate the runtime performance of our automated repair flow, we made a runtime comparison between the baseline job and corresponding repair job for a typical production layout. In Co-Opt repair OPC, the runtime depends on multiple factors such as defect location, pattern density, and computing resources. The recommended computing resources for tachyon FMO is one leaf node for every four defects to repair.

Table 3 shows that with sufficient computing resources, the runtime of Co-Opt repair flow is less than 10% of the production job. If the core resource is not enough, the runtime could be much longer than the production jobs.

Table 3. Runtime comparison between Co-Opt repair flow and a normal production job

| | *Case 1* | *Case 2* | *Case 3* | *case 4* |
|---|---|---|---|---|
| Core resource | Enough cores | Enough cores | Not enough cores (8.5% of recommended) | Not enough cores (1.8% of recommended) |
| Repair Runtime compared to typical customer chip | 6% | 9% | 205% | 320% |

## 6. CONCLUSIONS

In this work we have demonstrated that a flow for automated OPC repair can be set up and successfully used to correct hotspots on a 2x-nm line/space layout by using a Co-Optimization OPC application. The Co-Optimization approach was able to solve a wide range of hotspots without specifically tuning the recipe to address particular geometries. By using the FMO approach to limit the application of Co-Opt OPC only to hotspot sites, a defect-free OPC result was obtained while limiting the additional OPC runtime to about 9% of the baseline runtime.

## REFERENCES


[1] Chua, Gek Soon, Zou, Yi, Wang, Wei-Long, Qiu, Jianhong, Pandey, Taksh, Baron, Stanislas, Kapasi, Sanjay, Dover, Russell, Zhang, Xiaolong, "Cost Effective Application of Advanced Computational Lithography Techniques Using Flexible Mask Optimization". ASMC (Advanced Semiconductor Manufacturing Conference), May 2013.

[2] Charlotte Beylier; Nicolas Martin; Vincent Farys; Franck Foussadier; Emek Yesilada; Frederic Robert; Stanislas Baron; Russell Dover; Hua-yu Liu, "Demonstration of an effective flexible mask optimization (FMO) flow". Proc. SPIE. 8326, Optical Microlithography XXV (2012)